

CLAIMS

1. A network processor comprising:
 - a command FIFO to store one or more commands;
 - a plurality of cores;
 - 5 a like plurality of same context command FIFOs, each of said plurality of same context command FIFOs coupled to a corresponding one of said first plurality of cores; and
 - a scheduler coupled to said command FIFO and to each of said first plurality of cores in said core group.
- 10 2. The network processor of Claim 1 wherein said plurality of cores corresponds to a first plurality of cores in a first core group and the network processor further comprises a plurality of core groups, each of said core groups coupled to the command FIFO.
- 15 3. The network processor of Claim 1 wherein said plurality of cores corresponds to a first plurality of cores in a first core group and the network processor further comprises:
 - a plurality of core groups; and
 - said command FIFO corresponds to a first one of a plurality of command FIFOs, each of said plurality of command FIFOs coupled to a corresponding one of said plurality of core groups.
- 20 4. The network processor of Claim 3 wherein said plurality of same context command FIFOs is provided such that each of the cores in said plurality of core groups is coupled to a corresponding one of said plurality of same-context-command FIFOs.
- 25 5. The network processor of Claim 1 wherein said scheduler is adapted to receive a command from said command FIFO and to assign the command to one of said plurality of cores.
6. The network processor of Claim 1 wherein said scheduler is adapted to receive a

command from said command FIFO and to determine which of said plurality of cores is processing a command and in response to one of said plurality of cores in said core group processing a command, to determine if the command received from said command FIFO is in the same context as the command being processed by said core.

5

7. A network comprising:

a first node having a network processor, said network processor comprising:

a command FIFO to store one or more commands;

a plurality of cores;

10

a like plurality of same context command FIFOs, each of said plurality of same context command FIFOs coupled to a corresponding one of said first plurality of cores; and

a scheduler coupled to said command FIFO and to each of said first plurality of cores in said core group; and

15

an interface, adapted to couple the first node to another node.

8. The network of Claim 7 further comprising:

a network communication path coupled to said first node;

a second node having a network processor, said network processor comprising:

20

a command FIFO to store one or more commands;

a plurality of cores;

a like plurality of same context command FIFOs, each of said plurality of same context command FIFOs coupled to a corresponding one of said first plurality of cores; and

25

a scheduler coupled to said command FIFO and to each of said first plurality of cores in said core group; and

an interface coupled to said first node through said network communication path.

30

9. A processor comprising:

(a) a plurality of cryptographic acceleration units, each of said plurality of cryptographic acceleration units comprising:

(1) a command first-in-first-out (FIFO) queue to store one or more commands;

(2) a plurality of cores;

5 (3) a like plurality of same-context command FIFO queues, each of said plurality of same context command FIFO queues coupled to a corresponding one of said first plurality of cores; and

(4) a scheduler coupled to said command FIFO queue and to each of said first plurality of cores in said core group.

10

10. The processor of claim 9 further comprising:

a global FIFO queue having an input adapted to receive commands directed toward at least one of said plurality of cryptographic acceleration units and having an output; and

a global scheduler having an input coupled to the output of said global FIFO queue and

15 having an output adapted to provide a data path to each of said plurality of cryptographic acceleration units.

11. The processor of claim 10 wherein said plurality of cores form a first core group and wherein the processor further comprises a plurality of core groups, each of plurality of core groups coupled to said global scheduler.

20

12. A processor comprising:

a cryptographic acceleration unit comprising:

(1) a command first-in-first-out (FIFO) queue to store one or more commands;

25 (2) a core group comprising a plurality of cores;

(3) a like plurality of same-context command FIFO queues, each of said plurality of same context command FIFO queues coupled to a corresponding one of said plurality of cores; and

(4) a scheduler coupled to said command FIFO queue and to each of said first plurality of cores in said core group.

30

13. The processor of claim 12 wherein:

said core group corresponds to a first one of a plurality of core groups in said cryptographic acceleration unit, with each of said plurality of core groups comprising one or
5 more cores; and

said scheduler is coupled to each of said cores.

14. The processor of claim 13 wherein said cryptographic acceleration unit is a first one of a plurality of cryptographic acceleration units with each of said plurality of cryptographic
10 acceleration unit comprising a plurality of core groups with each of said plurality of core groups comprising one or more cores and a scheduler coupled to each of said cores and the processor further comprises:

a cross unit scheduler adapted to receive commands and to direct the commands to at least one of said plurality of cryptographic acceleration units.

15. A method comprising:

receiving a command in a command FIFO;

determining whether one of a plurality of cores is idle;

assigning the command to the idle one of the plurality of cores.

16. The method of claim 15 further comprising:

in response to all of the cores in the core group being idle, assigning the command in the command FIFO to a first pre-selected one of the plurality of cores;

in response to not all of the plurality of cores being idle, determining whether the
25 context of the command in the command FIFO is the same as the context of the command currently being processed by one of the plurality of cores which is not idle.

17. The method of claim 16 further comprising:

in response to the contexts being the same, moving the command from the command
30 FIFO to a same context command buffer associated with the core which is currently processing

the command in the same context.

18. The method of claim 16 further comprising:

in response to the context of the command in the command buffer not being the same as
5 the context of the command being processed by the core, assigning the command in the
command FIFO to a first one of the plurality of cores which is idle.

19. A method comprising:

receiving an indication that at least one core in a core group is idle;

10 determining if there are any commands in a command buffer;

in response to finding a command in the command FIFO, assigning the command to an
idle one of the plurality of cores in the core group.

20. The method of Claim 19, wherein prior to assigning the command to an idle one of the
15 plurality of cores in the core group, the method comprises determining whether any other core
in the core group is not idle.

21. The method of Claim 20 further comprising in response to all of the cores in the core
group being idle, assigning the command in the command FIFO to a first pre-determined one of
20 the plurality of cores in the core group.

22. The method of Claim 20 further comprising in response to not all of the cores in the
core group being idle, determining whether the context of the command in the command FIFO
is the same as the context of any command currently being processed by one of the cores which
25 is not idle.

23. The method of Claim 22 further comprising in response to the contexts being the same,
moving the command from the command FIFO to a same context command FIFO of the core
which is currently processing the command.

24. The method of Claim 22 further comprising in response to the context of the command in the command FIFO not being the same as the context of any command being processed by a core, assigning the command in the command FIFO to a first one of the plurality of cores which is idle.

5

25. An article comprising:

a storage medium having stored thereon instructions that when executed by a machine result in the following:

10 scheduling a packet command for processing by one of a plurality of cores depending upon whether a core is idle and whether the packet command has the same context of a command currently being processed by another one of the plurality of cores; and

if the packet command has the same context of a command currently being processed by another one of the plurality of cores, storing the command in a same-context command FIFO associated with the one of the plurality of cores.

15

26. The article of claim 25 wherein the storage medium has stored thereon instructions that when executed by a machine result moving the command from the same-context command FIFO to the core for processing.